DroMOOC

Trajectory planning

Advanced Level

Bézier and B-Spline curves: Example

Gauthier ROUSSEAU

L2S/CentraleSupélec

Introduction

Goal of this presentation

- Illustrate the use of the main properties of B-splines through a **simplistic** example
- Generation of a **feasible**, **collision-free** trajectory

Problem presentation

Problem statement

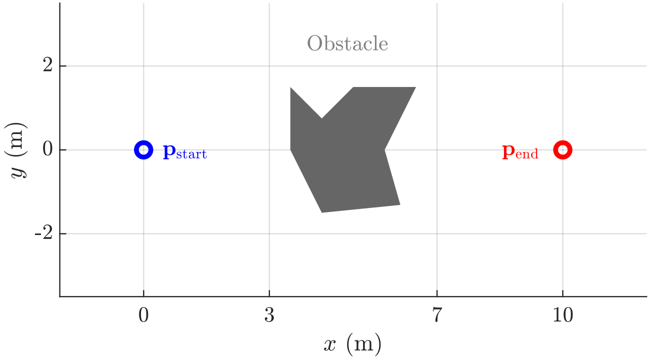
Considering a quadrotor initially at rest at $\mathbf{p}_{\mathrm{start}}$, generate trajectory ζ of duration T such that

- The quadrotor ends resting at $\mathbf{p}_{\mathrm{end}}$
- The trajectory is **collision-free**
- The ground speed of the quadrotor v does not exceed $v_{\max} = 2\mathrm{m.s}^{-1}$
- The angle of the quadrotor relatively to the ground α does not exceed $\alpha_{\max} = 15^\circ$
- The rotation speed ω of the quadrotor does not exceed $\omega_{\max} = 100^\circ.\mathrm{s}^{-1}$

Hypothesis

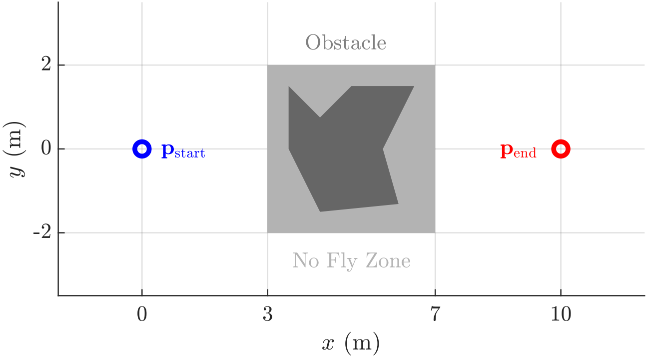
- 2D problem (constant altitude)
- No movement on the yaw axis

Problem presentation



Problem presentation

Obstacle bounded in a **convex no fly zone** (NFZ), with security margins



Strategy

B-spline trajectory generation in 2 steps

- **Control points.** Choose the control points such that the **path** is smooth and collision-free, using the convex hull property
- **Knot vector.** Choose the duration of the **trajectory** so that it is feasible, by applying the convex hull property on the control points of its derivatives

Use clamped, uniform B-splines as they are easy to work with

Uniform B-spline

Uniform B-spline

A B-spline is said **uniform** when its knots are equally distributed

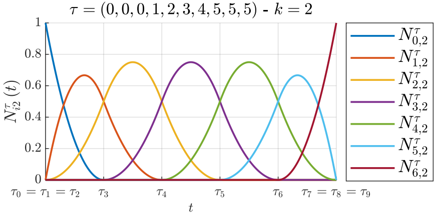
Knot vector replaced by 1 parameter, the step $\Delta\tau$ between 2 knots (and the first knot if $\neq 0$)

$$\boldsymbol{\tau} = \begin{pmatrix} \tau_0 & \tau_0 + \Delta\tau & \tau_0 + 2\,\Delta\tau & \dots & \tau_0 + m\,\Delta\tau \end{pmatrix}$$

Clamped B-splines

If the first $k+1$ knots are equal and the $k+1$ last knots are equal as well, then the B-spline is "clamped" to its first and last control points

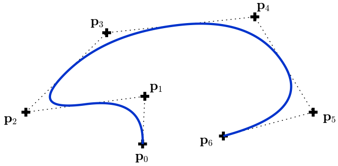
$$\boldsymbol{\tau} = \begin{pmatrix}\tau_0 & \tau_1 & \dots & \tau_m\end{pmatrix} = \big(\underbrace{\tau_k, \dots, \tau_k}_{k+1 \text{ knots}}, \underbrace{\tau_{k+1} \dots, \tau_n}_{n-k \text{ knots}}, \underbrace{\tau_{n+1}, \dots, \tau_{n+1}}_{k+1 \text{ knots}}\big)$$



"Uniform clamped" B-spline

If $\tau_0 = 0$ and the internal knots are equally distributed

$$\boldsymbol{\tau} = (\underbrace{0 \dots 0}_{k+1 \text{ knots}} \ \underbrace{\Delta\tau \dots (n-k)\Delta\tau}_{n-k \text{ knots}} \ \underbrace{(n-k+1)\Delta\tau \dots (n-k+1)\Delta\tau}_{k+1 \text{ knots}})$$



Clamped B-spline curves

Derivative

The derivative of a B-splines curve of degree $k > 0$ **is a also a B-spline curve**, with the same knots, a degree $k-1$ and $n+1$ control points given by a linear combination of the original ones

$$\mathcal{B}'_{\mathbf{P},\boldsymbol{\tau}} = \mathcal{B}_{\mathbf{P}^{(1)},\boldsymbol{\tau}} \quad \text{with} \quad \begin{cases} \mathbf{p}_0^{(1)} = \dfrac{k}{\tau_k - \tau_0}\mathbf{p}_0 \\ \forall i \in [\![1, n]\!] \;\; \mathbf{p}_i^{(1)} = \dfrac{k}{\tau_{i+k} - \tau_i}(\mathbf{p}_i - \mathbf{p}_{i-1}) \\ \mathbf{p}_{n+1}^{(1)} = -\dfrac{k}{\tau_{n+k+1} - \tau_{n+1}}\mathbf{p}_n \end{cases}$$

Clamped B-spline curves

Derivative

The derivative of a clamped B-splines curve of degree $k > 0$ **is a also a clamped B-spline curve** of degree $k-1$

- With n control points given by a linear combination of the original ones
- The same knot vector as the original clamped B-spline curve but with the multiplicity of the first and the last knots decreased by one

$$\mathcal{B}'_{\mathbf{P},\boldsymbol{\tau}} = \mathcal{B}_{\mathbf{P}^{(1)},\boldsymbol{\tau}^{(1)}} \tag{1}$$

with $$\begin{cases} \forall i \in [\![0, n-1]\!] \;\; \mathbf{p}_i^{(1)} = \dfrac{k}{\tau_{i+k+1} - \tau_{i+1}}(\mathbf{p}_{i+1} - \mathbf{p}_i) \\ \boldsymbol{\tau}^{(1)} = (\underbrace{\tau_k, \;\dots\;, \tau_k,}_{k \text{ knots}} \underbrace{\tau_{k+1} \;\dots\;, \;\tau_n,}_{n-k \text{ knots}} \underbrace{\tau_{n+1}, \;\dots\;, \;\tau_{n+1}}_{k \text{ knots}}) \end{cases}$$

Path generation

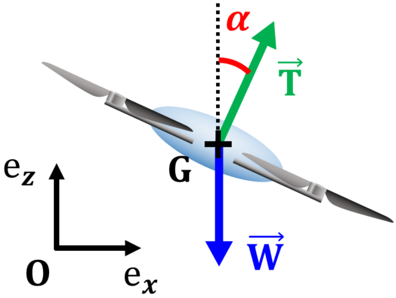
One way to generate a path: searching the **shortest** (in terms of **length**) **collision-free** B-spline curve joining the starting and the ending positions

3 parameters

- Degree k $\rightarrow$ differentiability class $\leqslant \mathcal{C}^{k-1}$
- Control Points $\mathbf{P}$ $\rightarrow$ collision free curve
- Knots $\boldsymbol{\tau}$ $\rightarrow$ uniform clamped B-spline

Path generation - Continuity

(Very) Simplistic 2D drone model: thrust and weight only



$$\overrightarrow{OP} = x\mathbf{e}_x + z\mathbf{e}_z$$

$$\begin{cases} \ddot{z} = -g + \dfrac{t}{m}\cos(\alpha) \\ \ddot{x} = \dfrac{t}{m}\sin(\alpha) \end{cases}$$

$$\ddot{z} = 0 \Rightarrow t = \frac{m\,g}{\cos(\alpha)}$$

$$\ddot{x} = g\tan(\alpha)$$

Path generation - Continuity

Rotation speed limited $\rightarrow$ drone attitude continuous

$$\alpha = \text{arctan}\left(\frac{\ddot{x}}{g}\right)$$

If the trajectory ζ is $\mathcal{C}^2$ then α is continuous

If all internal knots have a unit multiplicity, ζ is $\mathcal{C}^{k-1}$

$$k = 3$$

Path generation - Continuity

- A clamped B-spline starts on its first control point and ends on the last one
- The derivative of a clamped B-spline is a clamped B-spline

$\mathcal{C}^2$ rest-to-rest trajectory

$$\mathbf{p}_0 = \mathbf{p}_{\mathrm{start}}$$
$$\mathbf{p}_n = \mathbf{p}_{\mathrm{end}}$$
$$\mathbf{p}_0^{(1)} = \mathbf{p}_0^{(2)} = \mathbf{p}_{n-1}^{(1)} = \mathbf{p}_{n-2}^{(2)} = \mathbf{0}$$

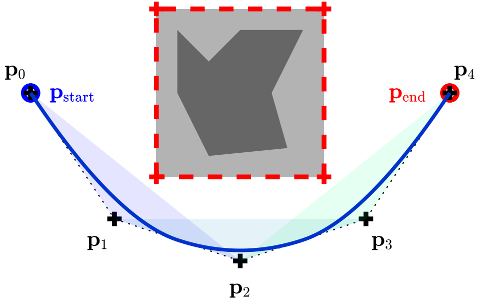
Using (1)

$$\mathbf{p}_0 = \mathbf{p}_1 = \mathbf{p}_2 = \mathbf{p}_{\mathrm{start}}$$
$$\mathbf{p}_n = \mathbf{p}_{n-1} = \mathbf{p}_{n-2} = \mathbf{p}_{\mathrm{end}}$$

Path generation - Obstacles management

Use convex hull property to garantee the absence of collision

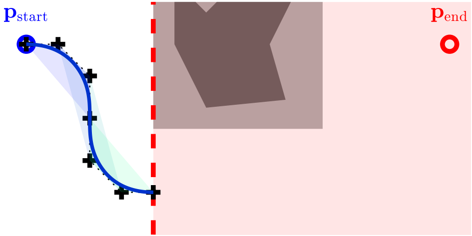
$\rightarrow$ Forbid the convex hulls to contain any vertices of the convex NFZ



Path generation - Obstacles management

This constraint can be hard to check

$\rightarrow$ simpler formulation with convex obstacle-free regions (more conservative)

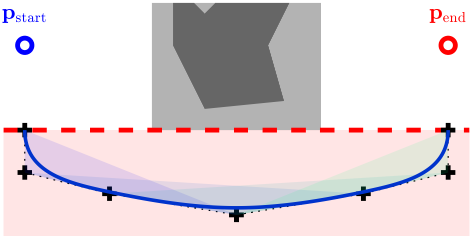


Region 1 $\quad p_i^x \leqslant 3$

Path generation - Obstacles management

This constraint can be hard to check

$\rightarrow$ simpler formulation with convex obstacle-free regions (more conservative)

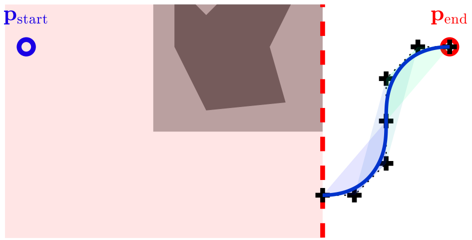


Region 2 $\quad p_i^y \leqslant -2$

Path generation - Obstacles management

This constraint can be hard to check

$\rightarrow$ simpler formulation with convex obstacle-free regions (more conservative)

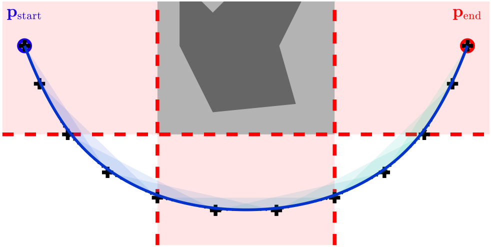


Region 3 $\quad -p_i^x \leqslant -7$

Path generation - Obstacles management

Some control points are in the convex hulls of other control points in different convex, obstacle-free regions

$\rightarrow$ These points are constrained in both regions



Region 1 & 2 $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} p_i^x \\ p_i^y \end{pmatrix} \leqslant \begin{pmatrix} 3 \\ -2 \end{pmatrix}$

Region 2 & 3 $\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} p_i^x \\ p_i^y \end{pmatrix} \leqslant \begin{pmatrix} -2 \\ -7 \end{pmatrix}$

Path generation - Number of control points

- 3 control points on the starting position
- n_1 control points in the first obstacle-free region
- $k = 3$ control points in both the first and second obstacle-free region
- n_2 control points in the second obstacle-free region
- $k = 3$ control points in both the second and third obstacle-free region
- n_3 control points in the third obstacle-free region
- 3 control points on the ending position

$$n + 1 = 6 + 2\,k + n_1 + n_2 + n_3$$

For $n_1 = n_2 = n_3 = 0$,

$$n = 11$$

Path generation - Knot vector

Only looking for a **clamped, uniform** B-spline **path** for now

$$\boldsymbol{\tau} = (\underbrace{0 \ \dots \ 0}_{k+1 \text{ knots}} \ \underbrace{\Delta\tau \ \dots \ (n-k)\Delta\tau}_{n-k \text{ knots}} \ \underbrace{(n-k+1)\Delta\tau \ \dots \ (n-k+1)\Delta\tau}_{k+1 \text{ knots}})$$

Fixed knot vector with **arbitrary step** $\Delta\tau$

$\boldsymbol{\tau} = (0, 0, 0, 0, 1, 2, \dots, 8, 9, 9, 9, 9)$

Path generation - Parameters and constraints

For a path of differentiability class $\mathcal{C}^L$

B-spline parameters

- $k = L + 1$
- $n = 2\,(L+1) + 2\,k + n_1 + n_2 + n_3 - 1$
- $\boldsymbol{\tau} = (\underbrace{0\ \dots\ 0}_{k+1 \text{ knots}}\ \underbrace{\Delta\tau\ \dots\ (n-k)\Delta\tau}_{n-k \text{ knots}}\ \underbrace{(n-k+1)\Delta\tau\ \dots\ (n-k+1)\Delta\tau}_{k+1 \text{ knots}})$

B-spline constraints

- $\forall i \in [\![0, L]\!]\ \ \mathbf{p}_i = \mathbf{p}_{\text{start}}$
- $\forall i \in [\![0, L]\!]\ \ \mathbf{p}_{n-i} = \mathbf{p}_{\text{end}}$
- $\forall i \in [\![L+1, L+n_1+k]\!]\ \ p_i^x \leqslant 3$
- $\forall i \in [\![L+n_1+1, L+n_1+n_2+2\,k]\!]\ \ p_i^y \leqslant -2$
- $\forall i \in [\![L+n_1+n_2+k+1, n-L-1]\!]\ \ -p_i^x \leqslant -7$

Path generation - Parameters and constraints

For a path of differentiability class $\mathcal{C}^2$, with $n_1 = n_2 = n_3 = 0$

B-spline parameters

- $k = 3$
- $n = 11$
- $\boldsymbol{\tau} = (0, 0, 0, 0, 1, 2, \ldots, 8, 9, 9, 9, 9)$

B-spline constraints

- $\mathbf{p}_0 = \mathbf{p}_1 = \mathbf{p}_2 = \mathbf{p}_{\text{start}}$
- $\mathbf{p}_n = \mathbf{p}_{n-1} = \mathbf{p}_{n-2} = \mathbf{p}_{\text{end}}$
- $\forall i \in [\![3, 5]\!] \;\; p_i^x \leqslant 3$
- $\forall i \in [\![3, 8]\!] \;\; p_i^y \leqslant -2$
- $\forall i \in [\![6, 8]\!] \;\; -p_i^x \leqslant -7$

Path generation - Criterion

Infinity of paths verifying the constraints

$\rightarrow$ Choose the best path according to a criterion

$\rightarrow$ Reduce the **length** of the path

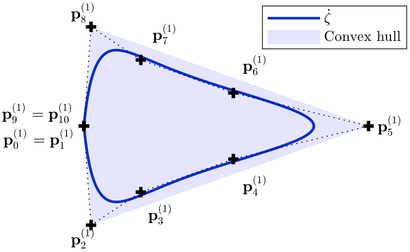
Length $\mathcal{L}$ of the curve ζ given by

$$\mathcal{L} = \int_0^{n-k+1} \left\| \dot{\zeta}(u) \right\|_2 \mathrm{d}u$$

$\dot{\zeta}$ lies in the convex hull of its control points

$\rightarrow$ Minimize the squared norm of its control points

$$J_1 = \sum_{i=0}^{n-1} \left\| \mathbf{p}_i^{(1)} \right\|_2^2$$



Path generation - Optimization problem

$$\mathbf{P}^* = \arg \min_{\mathbf{P} \in (\mathbb{R}^2)^{12}} \sum_{i=0}^{n-1} \left\| \mathbf{p}_i^{(1)} \right\|_2^2$$

$$\text{s.t.} \begin{cases} p_0^x = p_1^x = p_2^x = p_{\text{start}}^x \\ p_0^y = p_1^y = p_2^y = p_{\text{start}}^y \\ p_n^x = p_{n-1}^x = p_{n-2}^x = p_{\text{end}}^x \\ p_n^y = p_{n-1}^y = p_{n-2}^y = p_{\text{end}}^y \\ \forall i \in [\![3, 5]\!] \quad p_i^x \leqslant 3 \\ \forall i \in [\![3, 8]\!] \quad p_i^y \leqslant -2 \\ \forall i \in [\![6, 8]\!] \quad -p_i^x \leqslant -7 \end{cases} \tag{2}$$

Path generation - Optimization problem

Let be

$$\mathbf{x} = \begin{pmatrix} p_0^x & p_1^x & \dots & p_n^x & p_0^y & p_1^y & \dots & p_n^y \end{pmatrix}^\top$$

Then $J_1 = \mathbf{x}^\top \mathbf{H}_1 \mathbf{x}$, with $\mathbf{H}_1 = \begin{pmatrix} \frac{1}{\Delta\tau}\mathbf{Q}_1 & 0 \\ 0 & \frac{1}{\Delta\tau}\mathbf{Q}_1 \end{pmatrix}^\top \begin{pmatrix} \frac{1}{\Delta\tau}\mathbf{Q}_1 & 0 \\ 0 & \frac{1}{\Delta\tau}\mathbf{Q}_1 \end{pmatrix}$ and $\mathbf{Q}_1$ given by (1)

$$\mathbf{Q}_1 = \begin{pmatrix}
-3 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1.5 & 1.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.5 & 1.5 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3 & 3
\end{pmatrix}$$

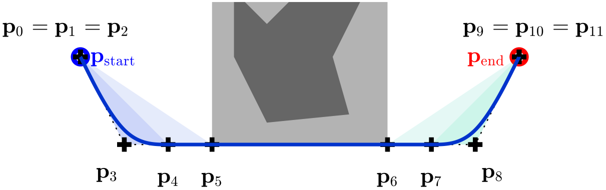
Path generation - Optimization problem

Problem (2) is a simple, small, convex QP problem

$$\mathbf{x}^* = \arg\min_{\mathbf{P}\in\mathbb{R}^{24}} \mathbf{x}^\top \mathbf{H}_1 \mathbf{x}$$
$$\text{s.t.} \begin{cases} A_{\text{eq}}\,\mathbf{x} = b_{\text{eq}} \\ A_{\text{ineq}}\,\mathbf{x} \leqslant b_{\text{ineq}} \end{cases} \tag{3}$$

Path generation - Optimization problem

Result: **Collision-free path**



A bit sub-optimal (not the shortest path)
due to the simplified formulation

Path generation - Smoothing term

Smooth the path by adding a penalty on the jerk $J = \sigma J_1 + (1-\sigma) J_3$

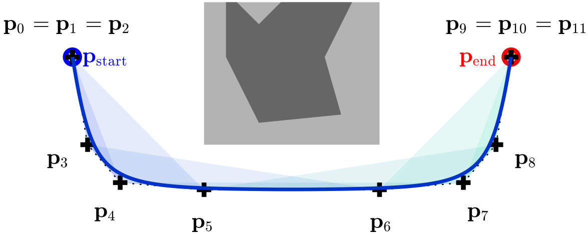
$$J_3 = \sum_{i=0}^{n-1} \left\| \mathbf{p}_i^{(3)} \right\|_2^2 = \mathbf{x}^\top \mathbf{H}_3 \mathbf{x}$$

with $\mathbf{H}_3 = \begin{pmatrix} \frac{1}{\Delta\tau^3}\mathbf{Q}_3 & 0 \\ 0 & \frac{1}{\Delta\tau^3}\mathbf{Q}_3 \end{pmatrix}^\top \begin{pmatrix} \frac{1}{\Delta\tau^3}\mathbf{Q}_3 & 0 \\ 0 & \frac{1}{\Delta\tau^3}\mathbf{Q}_3 \end{pmatrix}$ and $\mathbf{Q}_3$ given by (1)

$$\mathbf{Q}_3 = \begin{pmatrix} -6 & 10.5 & -5.5 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1.5 & 3.5 & -3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 3 & -3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 3 & -3 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 3 & -3 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 3 & -3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 3 & -3 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 3 & -3.5 & 1.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 5.5 & -10.5 & 6 \end{pmatrix}$$

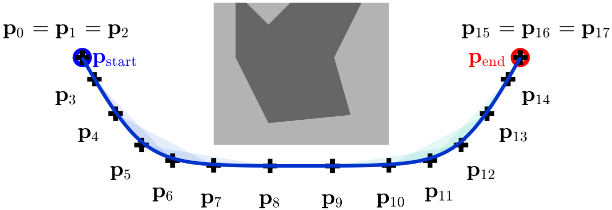
Path generation - Optimization problem

$\sigma = 0.8$



Path generation - Optimization problem

$n_1 = n_2 = n_3 = 2$, $\sigma = 0.5$

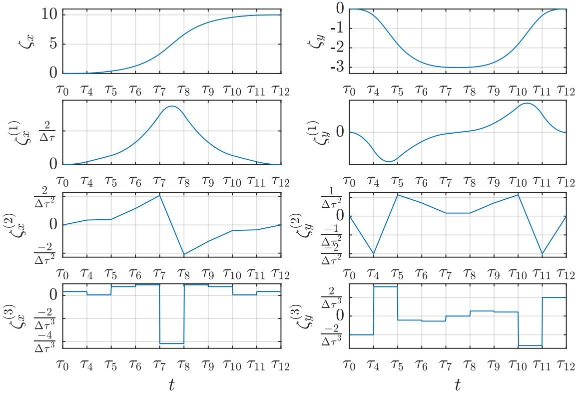


Trajectory duration

Transform the path into a trajectory by adding a duration

$\rightarrow$ Choice of $\Delta\tau$

Path Derivatives



Trajectory duration

Specification

- $v \leqslant v_{\max}$

Feasibility

- $\alpha \leqslant \alpha_{\max}$
- $\omega \leqslant \omega_{\max}$

Translate into constraints on the derivatives
$\rightarrow$ Then, use convex hull property

Trajectory duration

Acceleration

$a = g \tan(\alpha)$ with $a = \left\|\ddot{\zeta}\right\|_2$

$$(a \leqslant g \tan(\alpha_{\max})) \Rightarrow (\alpha \leqslant \alpha_{\max})$$

Jerk

$$\begin{cases} \ddot{\zeta} = \mathbf{R}\,\frac{t}{m}\mathbf{e}_z + \mathbf{g} \\ \dot{\mathbf{R}} = \mathbf{R}\,\hat{\omega} \end{cases}$$

$$t = \left\|\ddot{\zeta} - \mathbf{g}\right\|_2 = \frac{m\,g}{\cos(\alpha)}$$

$$\begin{cases} \dddot{\zeta} = \mathbf{R}\,\hat{\omega}\,\frac{t}{m}\mathbf{e}_z + \mathbf{R}\,\frac{\dot{t}}{m}\mathbf{e}_z \\ \dot{t} = m\,(\mathbf{R}\mathbf{e}_z)^\top \dddot{\zeta} \end{cases}$$

$$\begin{pmatrix} \omega_y \\ \omega_x \end{pmatrix} = \frac{\cos(\alpha)}{g} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix} \mathbf{R}^\top \dddot{\zeta}$$

By setting $\omega_z = 0$

$$(j \leqslant g\,\omega_{\max}) \Rightarrow (\omega \leqslant \omega_{\max})$$

Trajectory duration - Optimization problem

Minimum-time trajectory

$$
\Delta\tau^* = \arg\min_{\Delta\tau\in\mathbb{R}} \Delta\tau
$$

$$
\text{s.t.} \begin{cases} \Delta\tau > 0 \\ \forall i \in [\![0, n-1]\!] \quad \left\|\mathbf{p}_i^{(1)}\right\|_2 \leqslant v_{\max} \\ \forall i \in [\![0, n-2]\!] \quad \left\|\mathbf{p}_i^{(2)}\right\|_2 \leqslant g \tan\left(\alpha_{\max}\right) \\ \forall i \in [\![0, n-3]\!] \quad \left\|\mathbf{p}_i^{(3)}\right\|_2 \leqslant m\, g\, \omega_{\max} \end{cases} \tag{4}
$$

Trajectory duration - Optimization problem

$$\mathbf{P}^{(1)} = \mathbf{P}\,\frac{1}{\Delta\tau}{\mathbf{Q}_1}^\top = \frac{1}{\Delta\tau}\,\widetilde{\mathbf{P}}^{(1)}$$

$$\mathbf{P}^{(2)} = \mathbf{P}\,\frac{1}{\Delta\tau^2}{\mathbf{Q}_2}^\top = \frac{1}{\Delta\tau^2}\,\widetilde{\mathbf{P}}^{(2)}$$

$$\mathbf{P}^{(3)} = \mathbf{P}\,\frac{1}{\Delta\tau^3}{\mathbf{Q}_3}^\top = \frac{1}{\Delta\tau^3}\,\widetilde{\mathbf{P}}^{(3)}$$

with $\widetilde{\mathbf{P}}^{(l)} = \mathbf{P}\,{\mathbf{Q}_l}^\top$, fixed, and

$$\mathbf{Q}_2 = \begin{pmatrix}
6 & -9 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1.5 & -2.5 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2.5 & 1.5 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & -9 & 6
\end{pmatrix}$$

Trajectory duration - Optimization problem

(4) can be written

$$\Delta\tau^* = \arg\min_{\Delta\tau\in\mathbb{R}} \Delta\tau$$

$$\text{s.t.} \begin{cases} \Delta\tau \geqslant \varepsilon \\ \forall i \in [\![0, n-1]\!] \;\; \Delta\tau \geqslant \dfrac{\left\|\widetilde{\mathbf{p}}_i^{(1)}\right\|_2}{v_{\max}} \\ \forall i \in [\![0, n-2]\!] \;\; \Delta\tau \geqslant \sqrt{\dfrac{\left\|\widetilde{\mathbf{p}}_i^{(2)}\right\|_2}{g \tan\left(\alpha_{\max}\right)}} \\ \forall i \in [\![0, n-3]\!] \;\; \Delta\tau \geqslant \sqrt[3]{\dfrac{\left\|\widetilde{\mathbf{p}}_i^{(3)}\right\|_2}{m\, g\, \omega_{\max}}} \end{cases}$$

Trajectory duration - Optimization problem

$$\Delta\tau^* = \arg\min_{\Delta\tau\in\mathbb{R}} \Delta\tau$$
$$\text{s.t.}\ \ \mathbf{1}_{3n-2}\,\Delta\tau \geqslant \boldsymbol{\Delta\tau}_{\min}$$

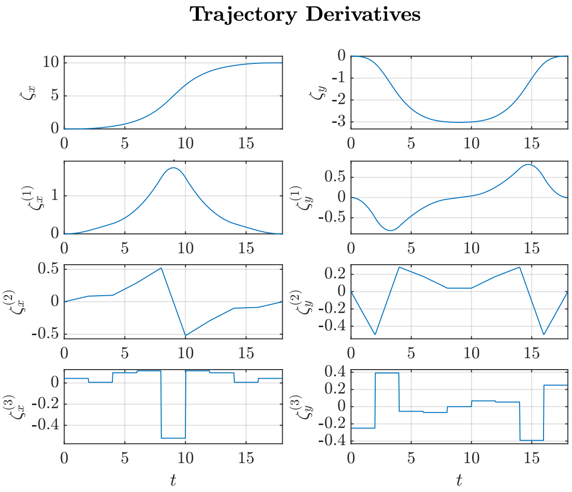
with $\mathbf{1}_i = \underbrace{\begin{pmatrix}1 & 1 & \dots & 1\end{pmatrix}}_{i\text{ elements}}^{\top}$

The solution is trivial

$$\Delta\tau^* = \max\left(\boldsymbol{\Delta\tau}_{\min}\right)$$

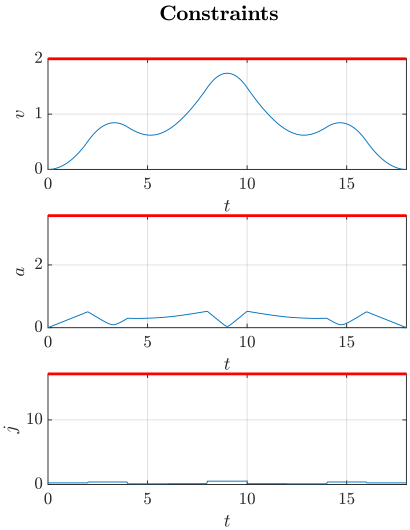
Trajectory duration - Optimization problem

$n_1 = n_2 = n_3 = 0$, $\sigma = 0.8$



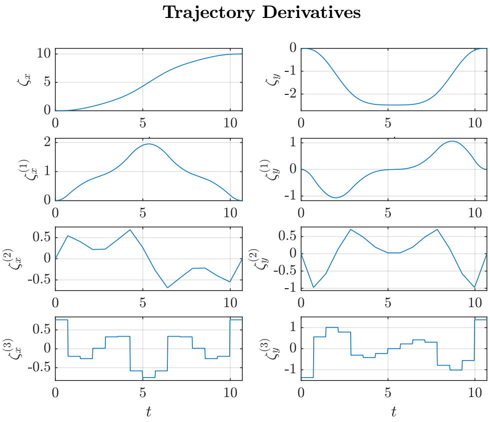
Trajectory duration - Optimization problem

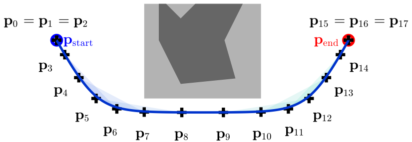
$n_1 = n_2 = n_3 = 0$, $\sigma = 0.8$



Trajectory duration - Optimization problem

$n_1 = n_2 = n_3 = 2$, $\sigma = 0.5$





Conclusion

- Simple trajectory generator satisfying the specifications
- Extremely fast computation ($< \mathrm{ms}$)

- Conservative
- Simplistic (2D, no disturbances, known convex NFZ, simplified model etc.)